

CLAIMS

What is claimed is:

- 1 1. A method for programmable pixel processing in a computer graphics
2 pipeline, comprising:
3 (a) receiving pixel data from a source buffer;
4 (b) performing programmable operations on the pixel data in order to generate
5 output, wherein the operations are programmable by a user utilizing
6 instructions from a predetermined instruction set; and
7 (c) storing the output in a register.
- 1 2. The method as recited in claim 1, wherein the output stored in the register is
2 used in performing the programmable operations on the pixel data.
- 1 3. The method as recited in claim 1, wherein the pixel data includes a position,
2 a pixel diffuse color, a specular color, a fog value, and a plurality of texture
3 coordinates.
- 1 4. The method as recited in claim 1, wherein the pixel data is selected from the
2 group consisting of a position, a pixel diffuse color, a specular color, a fog
3 value, and a plurality of texture coordinates.
- 1 5. The method as recited in claim 1, and further comprising performing an
2 operation involving the output, the operation selected from the group
3 consisting of a scissor operation, a color format conversion, an alpha test
4 operation, a z-buffer/stencil operation, a blend operation, a logic operation, a
5 dither operation, and a writemask operation.

- 1 6. The method as recited in claim 1, wherein further standard operations are
2 performed on the pixel data utilizing a standard graphics application program
3 interface (API).
- 1 7. The method as recited in 1, wherein the output includes a color value and a
2 depth value .
- 1 8. The method as recited in claim 1, and further comprising negating the pixel
2 data prior to performing the programmable operations thereon.
- 1 9. The method as recited in claim 1, and further comprising swizzling the pixel
2 data prior to performing the programmable operations thereon.
- 1 10. The method as recited in claim 1, wherein the programmable operations
2 includes a texture fetch operation.
- 1 11. The method as recited in claim 10, wherein the texture fetch operation
2 involves a slope.
- 1 12. The method as recited in claim 10, wherein the texture fetch operation is
2 capable of being used in a level of detail (LOD) calculation.
- 1 13. The method as recited in claim 1, wherein the programmable operations
2 support multiple levels of precision.
- 1 14. The method as recited in claim 13, wherein the levels of precision include
2 full floating point, half floating point, and fixed point.
- 1 15. The method as recited in claim 13, wherein the programmable operations are
2 capable of converting the pixel data from a first level of precision to a second
3 level of precision.

- 1 16. The method as recited in claim 1, wherein the programmable operations are
2 capable of clamping the pixel data for packing the pixel data into a
3 destination.
- 1 17. The method as recited in claim 1, wherein condition codes are initialized
2 prior to the programmable operations being performed.
- 1 18. The method as recited in claim 1, wherein the programmable operations are
2 capable of removing the pixel data.
- 1 19. The method as recited in claim 1, wherein the programmable operations are
2 selected from the group consisting of a no operation, texture fetch, move,
3 derivative, multiply, addition, multiply and addition, reciprocal, reciprocal
4 square root, three component dot product, four component dot product,
5 distance vector, minimum, maximum, pack, unpack, set on less than, set on
6 greater or equal than, floor, fraction, kill pixel, exponential base two (2),
7 logarithm base two (2), and light coefficients.
- 1 20. A computer program product for programmable pixel processing in a
2 computer graphics pipeline, comprising:
3 (a) computer code for receiving pixel data from a source buffer;
4 (b) computer code for performing programmable operations on the pixel data in
5 order to generate output, wherein the operations are programmable by a user
6 utilizing instructions from a predetermined instruction set; and
7 (c) computer code for storing the output in a register.
- 1 21. A system for programmable pixel processing, comprising:
2 (a) a source buffer for storing pixel data;
3 (b) a functional module coupled to the source buffer for performing
4 programmable operations on the pixel data received therefrom in order to

- 5 generate output, wherein the operations are programmable by a user utilizing
6 instructions from a predetermined instruction set; and
7 (c) a register coupled to the functional module for storing the output.

- 1 22. A method for programmable pixel processing in a computer graphics
2 pipeline, comprising:
3 (a) receiving pixel data from a source buffer;
4 (b) performing programmable operations on the pixel data in order to generate
5 output, wherein the operations are programmable by a user utilizing
6 instructions from a predetermined instruction set, and the programmable
7 operations support multiple levels of precision; and
8 (c) converting the pixel data from a first level of precision to a second level of
9 precision.

- 1 23. A method for programmable pixel processing in a computer graphics
2 pipeline, comprising:
3 (a) receiving pixel data from a source buffer;
4 (b) performing programmable operations on the pixel data including a texture
5 fetch in order to generate output, wherein the operations are programmable
6 by a user utilizing instructions from a predetermined instruction set; and
7 (c) storing the output in a register.

- 1 24. A method for programmable pixel processing in a computer graphics
2 pipeline, comprising:
3 (a) determining whether the graphics pipeline is operating in a programmable
4 mode;
5 (b) performing programmable operations on pixel data in order to generate
6 output if it is determined that the graphics pipeline is operating in the
7 programmable mode; and
8 (c) performing standard operations on the pixel data in order to generate output
9 in accordance with a standard graphics application program interface if it is

10 determined that the graphics pipeline is not operating in the programmable
11 mode.

1 25. The method as recited in claim 24, wherein the standard graphics application
2 program interface includes OpenGL[®].

1 26. A computer program product for programmable pixel processing in a
2 computer graphics pipeline, comprising:

3 (a) computer code for determining whether the graphics pipeline is operating in
4 a programmable mode;

5 (b) computer code for performing programmable operations on pixel data in
6 order to generate output if it is determined that the graphics pipeline is
7 operating in the programmable mode; and

8 (c) computer code for performing standard operations on the pixel data in order
9 to generate output in accordance with a standard graphics application
10 program interface if it is determined that the graphics pipeline is not
11 operating in the programmable mode.

1 27. The computer program product as recited in claim 26, wherein the standard
2 graphics application program interface includes OpenGL®.

1 28. A system for programmable pixel processing in a computer graphics
2 pipeline, comprising:

3 (a) means for determining whether the graphics pipeline is operating in a
4 programmable mode;

5 (b) means for performing programmable operations on pixel data in order to
6 generate output if it is determined that the graphics pipeline is operating in
7 the programmable mode; and

8 (c) means for performing standard operations on the pixel data in order to
9 generate output in accordance with a standard graphics application program

10 interface if it is determined that the graphics pipeline is not operating in the
11 programmable mode.

1 29. The computer program product as recited in claim 29, wherein the standard
2 graphics application program interface includes OpenGL®.

1 30. A method for programmable pixel processing in a computer graphics
2 pipeline, comprising:

3 (a) determining whether the graphics pipeline is operating in a programmable
4 mode;

5 (b) performing programmable operations on pixel data in order to generate
6 output if it is determined that the graphics pipeline is operating in the
7 programmable mode; and

8 (c) performing standard operations on the pixel data in order to generate output
9 in accordance with a standard graphics application program interface if it is
10 determined that the graphics pipeline is not operating in the programmable
11 mode;

12 (d) wherein the programmable operations are selected from the group consisting
13 of a no operation, texture fetch, move, derivative, multiply, addition,
14 multiply and addition, reciprocal, reciprocal square root, three component dot
15 product, four component dot product, distance vector, minimum, maximum,
16 pack, unpack, set on less than, set on greater or equal than, floor, fraction,
17 kill pixel, exponential base two (2), logarithm base two (2), and light
18 coefficients

add
cu